Adatbiztonság pótpótZH Név:
2014. május 26. Neptun kód:

1. Hash függvény:

a.) Definiálja az iterációs hash függvényt! **(2p)**
b.) Nehéz feladat-e H(m, H0) = H(m*, H0*), m≠m* pszeudo ütközést előállítani? (Indoklásul algoritmust, vagy bizonyítást adjon!) **(3p)**
c.) Változik-e az b.) kérdésre adott válasza, ha DM paddinget is alkalmaz a támadott tömörítő eljárás? **(3p)**

2. RSA:

a.) Igaz-e, hogy az RSA rejtjelezés nem biztonságos olyan támadással szemben, amelyben a támadó azt tűzi ki célul, hogy megfejt egy y rejtett RSA blokkot úgy, hogy mindössze egy kérést küldhet egy dekódoló orákulumhoz, amely orákulum egy tetszőleges, y*≠y blokkot dekódol a számára? Indokoljon! **(4 p)**
b.) Igaz-e, hogy aláíró orákulumhoz fordulás nélkül tudunk érvényes RSA (üzenet, aláírás) párt produkálni? Indokoljon! **(4p)**

3. Jelszavas hitelesítés: Melyik biztonságosabb? Számítással indokoljon!

i) (a) egy 4 alfanumerikus (case sensitive) karakterből álló véletlen jelszó vagy (b) egy 8 számjegyből álló véletlen PIN kód? **(3p)**
ii) (a) egy 6 számjegyből álló véletlen PIN kód vagy (b) egy 8 karakterből álló átlagos felhasználó által választott átlagos jelszó? **(3p)**

4. WEP:

a.) Röviden írja le, hogyan támadható a WEP protokollban az integritásvédelem céljára használt konstrukció! **(2p)**
b.) A WEP integritásvédelmét úgy próbáljuk kijavítani, hogy a konstrukcióban használt CRC ellenőrző-összeg számítás helyett egy új ellenőrző-összeg számítást vezetünk be az alábbiak szerint: az üzenetet 128 bites blokkokra osztjuk, ha az utolsó blokk rövidebb 128 bitnél, akkor 0 bitekkel 128 bitre egészítjük ki, majd az így kapott blokkok (bitenkénti) XOR összegét számolva kapjuk az ellenőrző-összeget. A konstrukció minden más elemét megtartjuk, többek között az RC4 rejtjelezőt is. Helyesen járunk-e el? Indokoljon! **(3p)**
c.) A WEP integritásvédelmét most úgy próbáljuk kijavítani, hogy a CRC ellenőrző-összeget megtartjuk, de RC4 helyett RC5 rejtjelezőt használunk CBC módban. Helyesen járunk-e el? Indokoljon! **(3p)**

5. Kis kérdések:

a.) Definiálja a DoS támadás fogalmát röviden! **(1p)**
b.) Mi az a „magic packet” jellegű DoS támadás? **(2p)**
c.) Illusztrálja rajzzal a DNS amplification támadás lényegét! **(2p)**

6. Unix/Linux hozzáférésvédelem az órán előadott módon. Tekintsük az alábbi `/etc/passwd` file részletet:

```
u1:x:1003:1004:,,,:/home/u1:/bin/bash
u2:x:1004:1005:,,,:/home/u2:/bin/bash
u3:x:1005:1006:,,,:/home/u3:/bin/bash
u4:x:1006:1007:,,,:/home/u4:/bin/bash
```

Az `/etc/group` file releváns része:

```
u1:x:1004:
u2:x:1005:
u3:x:1006:
u4:x:1007:
g1:x:1008:u1,u2
g2:x:1009:u2,u3,u4
g3:x:1010:u2,u3
```

A fájl hozzáférési jogosultságok az alábbiak:

```
root@gotcha:/adatbizt# ls -la
total 16
drwxr-xr-x  4 root root 4096 2011-04-22 10:49 .
drwxr-xr-x 25 root root 4096 2011-04-22 10:51 ..
drwxrwsr-x  2 u1   g1   4096 2011-04-22 10:50 d1
drwxr-xr-x  2 u2   g1   4096 2011-04-22 10:50 d2

root@gotcha:/adatbizt# ls -la d1
total 20
drwxrwsr-x 2 u1   g1   4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw------- 1 u1   u4      4 2011-04-22 10:50 f1
-rw-rw---- 1 u1   g1     16 2011-04-22 10:50 f2
-rwxrwxrwx 1 u1   g2      8 2011-04-22 10:50 f3

root@gotcha:/adatbizt# ls -la d2
total 16
drwxr-xr-x 2 u2   g1   4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw-r--r-- 1 root g1      7 2011-04-22 10:50 f4
--w------- 1 root g1      6 2011-04-22 10:50 f5
```

a.) Mely felhasználók tudják olvasni a `d1/f1` fájlt és miért? **(2p)**
b.) Mely felhasználóknál fut le sikeresen a `cp d1/f2 d2/f6` parancs? **(2p)**
c.) Ki tudja módosítani az `f2` fájl jogosultságait (pl. `chmod o+w d1/f2`)? **(3p)**
d.) Ki tudja lefuttatni az `rm d1/f3; cp d1/f1 d1/f3` parancsokat mind sikeresen, azaz kitörölni `f3`-at és helyére `f1`-et másolni? **(3p)**

Pontozás: 1: 0-18, 2: 19-25, 3: 26-32, 4: 33-39, 5: 40-45

Adatbiztonság pótpót ZH megoldások
2014. május 26.

1. Megoldás:
a.) ea. slide ill. tk. 418 old. 4.3 példa
b.) Nem nehéz. Támadás: H([m1,m2],H0) = f(m2,f(m1,H0)) = H(m2,H0*), ahol H0*=f(m,H0)
c.) Amennyiben ezt a támadást adta meg, ez nem működik MD padding mellett, mivel H([m1,m2, hossz_1], H0) $\neq$ H([m2, hossz_2], H0*).

2. Megoldás:
a.) Igaz. Vak-aláírás ötletét alkalmazza a támadó, $y^*=R^e y \pmod m$ blokkra kér dekódolást az orákulumtól, ahol R egy véletlen, mod m invertálható elem, majd mod m osztással eliminálja R elemet az orákulum válaszából.
b.) Igaz. ($v^e \bmod m$, v) , ahol $v < m$.

3. Megoldás:
i) (a) 4*log62 = 4*5.954 = 23.816 bit (b) 8*log10 = 8*3.322 = 26.656 bit
ii) (a) 6*log10 = 6*3.322 = 19.932 bit (b) 4 + 7*2 = 18 bit

4. Megoldás:
a) Az integritásvédelem a WEP-ben úgy működik, hogy kiszámoljuk az üzenet CRC ellenőrző-összegét, majd az üzenetet és a CRC-t együtt rejtjelezzük RC4 rejtjelezővel. A probléma, hogy a CRC lineáris az XOR műveletre nézve, és ezért a támadó teszőleges változást elő tud idézni a WEP üzenetben (előadás 14. fólia):

$$((M \mid CRC(M)) + K) + (\Delta M \mid CRC(\Delta M)) =$$
$$((M + \Delta M) \mid (CRC(M) + CRC(\Delta M))) + K =$$
$$((M + \Delta M) \mid CRC(M + \Delta M)) + K$$

ahol K az RC4 által generált kulcssorozat.

b) Nem, mert ez a konstrukció ugyanúgy lineáris az XOR műveletre nézve.

c) Igen, mert az RC5 CBC módban nem kulcsfolyamrejtjelezést valósít meg, és így a támadó nem tud célzott módosítást végezni az üzenet bitjein (ha egy bitet módosít a rejtett WEP üzenetben, nem tudja, hogy a nyílt üzenetben milyen bitek változnak meg ennek hatására).

5. Megoldás:
lásd előadás fóliák

6. Megoldás:
a.) csak u1, neki van olvasás joga és a directory-ba is be tud lépni
b.) u2, root, mert: g1 és u1 tudja olvasni f2-t, joguk van belépni d2-be, ott fájlt azonban u2 tud csak csinálni.
c.) csak a tulajdonos, u1 (és a root)
d.) a törlést csak u1,u2 tudja elvégezni f1-et viszont ebből csak u1 olvashatja f1-et, ő írni is tud az alkönyvtárba, tehát csak u1. (+root)